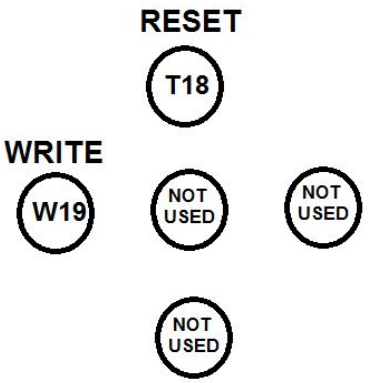
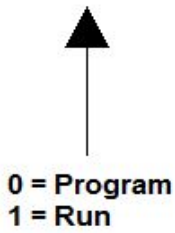


Memory Address		**When writing an instruction** **Bit 9 to 5**: op-code or instruction **Bit 4 to 0**: address being operated on
		When writing data to an address **Bit 9 to 0**: all used for the input



Op Code		
Mnemonic	**Binary word**	**Meaning**
LDA	00000	Load Register A
ADD	00001	Add to Register A
SUB	00010	Subtract from Register A
JMP	00011	Jump to specified address
CMP	00100	Compare A with value in B
JEQ	00101	Jump if A = B
JGR	00110	Jump if A > B
JLS	00111	Jump if A < B
MOV	01000	Move to specified address
OUT	01110	Output to display
HLT	11111	Halt

This program will add a number to itself (starting at 1) until it reaches 512 and start over again. The program loops back once it reaches 512 because adding 512 to itself would cause an overflow.

Description	Actual input to Basys Board		
	Memory Address	**Program**	**Instruction or Data**
Load register A with memory address 11111 LDA 11111	00000	0	00000 11111
Add value in memory address 11111 to Register A ADD 11111	00001	0	00001 11111
Output result of addition to the display OUT ------	00010	0	01110 - - - - -
Move result of the addition to memory address 11111 MOV 11111	00011	0	01000 11111
Compare Register A with the value in memory address 11110 CMP 11110	00100	0	00100 11110
If the values compared are equal, jump to memory address 00111 JEQ 00111	00101	0	00101 00111
If not, jump back to memory address 00000 JMP 00000	00110	0	00011 00000
Load register A with memory address 11100 LDA 11100	00111	0	00000 11100
Move result of the addition to memory address 11111 MOV 11111	01000	0	01000 11111
Jump back to memory address 00000 JMP 00000	01001	0	00011 00000
. . . .	. . . .	. . . .	. . . .
Store 1 in this location, this what is loaded into A when the loop reaches 512 00000 00001	11100	0	00000 00001
Store 512 in this location, this control or limit for the loop 10000 00000	11110	0	10000 00000
Store 1 in this location, this is where the result of the additions will be stored 00000 00001	11111	0	00000 00001